多模态多目标差分进化算法求解非线性方程组 *

许伟伟[1]，梁　静[1]，岳彩通[1]，瞿博阳[2]

(1. 郑州大学 电气工程学院，郑州 450001; 2. 中原工学院 电子信息学院，郑州 450007)

摘　要：针对当前算法在求解非线性方程组时面临解的个数不完整、精确度不高、收敛速度慢等问题进行了研究，提出一种多模态多目标差分进化算法。首先将非线性方程组转换为多模态多目标优化问题，初始化一个随机种群并对种群中全部个体进行评价;然后通过非支配解排序和决策空间拥挤距离选择机制,挑选种群中的一半优质个体进行变异;接着在变异过程中采用一种新的变异策略和边界处理方法以增加解的多样性；最后通过交叉和选择机制使优质个体进行进化，直到搜索到全部最优解。在所选测试函数集和工程实例上的实验结果表明，该算法能有效地搜索到非线性方程组的解，并通过与当前四个算法进行比较，该算法在解的数量和成功率上具有优越性。

关键词：非线性方程组；多模态；多目标；差分进化；非支配解排序

中图分类号：TP183　　doi: 10.3969/j.issn.1001-3695.2017.11.0734

Multimodal multi-objective differential evolution algorithm for solving nonlinear equations

Xu Weiwei[1], Liang Jing[1], Yue Caitong[1], Qu Boyang[2]

(1. *School of Electrical Engineering*, *Zhengzhou University*, *Zhengzhou* 450001, *China*; 2. *School of Electronic & Information Engineering*, *Zhongyuan University of Technology*, *Zhengzhou* 450007, *China*)

Abstract: Aiming at the problems such as incomplete solutions, low accuracy and slow convergence speed of current algorithms in solving nonlinear equations, this paper proposed a multimodal multi-objective differential evolution algorithm. Firstly, it transformed the nonlinear equations into multimodal multi-objective optimization problems, and initialized a random population and evaluated all individuals in the population. Through the non-dominated sorting and decision space crowding distance selection mechanism, it selected half of the individuals in the population to mutate. Then it used a new mutation strategy and boundary processing method to increase the diversity of solutions. Finally, it evolved the high-quality individuals through crossover and selection mechanism until it found all the optimal solutions. The experimental results on selected test function sets and engineering examples show that the algorithm can effectively search for optimal solutions and it is superior to the other four algorithms in the number of solutions and success rate.

Key words: nonlinear equations; multimodal; multi-objective; differential evolution; non-dominated sorting

0　引言

非线性方程组的求解一直以来都是数学学科中的一个难题。在实际生活中，很多物理、生物、工业里复杂的问题都可以看做是非线性方程组问题[1,2]。考虑到非线性方程组本身的复杂性，如何有效地获得非线性方程组的所有解已成为目前研究的主要目标。传统数学方法如牛顿法、拟牛顿法、欧拉法、复合形法、梯度法、迭代法、共轭方向法、混沌法、最速下降法、单纯型法等，大多对问题的初始值具有很严格的要求；另外对于某些导数不存在或是导数难求的方程，传统数学方法也具有一定的局限性。与传统数学方法相比，进化算法是一种成熟的具有高鲁棒性和广泛适用性的全局优化方法，具有自组织、自适应、自学习的特性，能够不受问题性质的限制，有效地处理传统数学方法难以解决的复杂问题。因此，利用进化算法求解非线性方程组已成为当前一个重要的研究方向。目前应用于求解非线性方程组的主要进化算法有遗传算法[3]、差分进化算法[4]、人工鱼群算法[5]、人工蜂群算法[6]、人工萤火虫算法[7]、布谷鸟算法[8]等。

差分进化算法作为当前流行的进化算法之一，由于其容易实现、可控参数少等特点，已经被一些研究学者用来求解非线

收稿日期：2017-11-04；**修回日期：**2018-01-04　**基金项目：**国家自然科学基金资助项目（61473266，61673404）；河南省高校优秀青年教师研究奖励基金资助项目（2014GGJS-004）；河南省大学创新人才科技计划（16HASTIT041）

作者简介：许伟伟（1991-），男，河南安阳人，硕士研究生，主要研究方向为智能计算、差分进化（xuweiwei888@163.com）；梁静（1981-），女，河南兰考人，教授，博士，主要研究方向为计算智能的理论研究及应用；岳彩通（1990-），男，博士研究生，主要研究方向为多目标优化、稀疏优化；瞿博阳（1984-），男，河南焦作人，副教授，博士，主要研究方向为智能优化算法，多目标优化.

性方程组。陈海霞等人[9]提出将非线性方程组转换为一个不可微优化问题，并用光滑函数代替不可微的极大值函数，最终结果表明该算法在求解非线性方程组的准确性和有效性上均优于其他同期算法。针对差分进化算法进化后期收敛缓慢和稳定性不强的缺陷，刘利斌等人[10]将拟牛顿法插入到差分进化算法当中，有效提高了差分进化算法的收敛速度和鲁棒性。封全喜等人[11]通过代理约束概念和修正极大熵函数，将非线性方程组转换为无约束优化函数并利用自适应正交差分进化算法求解，验证了所提算法的有效性。

上述算法在一定程度上实现了非线性方程组的求解，但仍然存在所得解的个数不完整、精确度不高、收敛速度慢等问题。为解决这些问题，本文提出了一种多模态多目标差分进化算法。首先将种群初始化并计算每个个体的适应度值，然后通过一种预选择机制将优质个体筛选进行变异，接着在变异过程中采用新的变异策略和边界处理办法提高搜索解的多样性，最后经过交叉选择优质个体进行多次进化，最终搜索到全部最优解。实验结果表明，该算法能够有效提高非线性方程组的解的数量和质量。

1 问题描述和转换方法

一般地，一个非线性方程组可以被描述如下：

$$\begin{cases} f_1(x_1,x_2,x_3,\dots,x_n)=0 \\ f_2(x_1,x_2,x_3,\dots,x_n)=0 \\ \qquad\vdots \\ f_m(x_1,x_2,x_3,\dots,x_n)=0 \end{cases} \tag{1}$$

其中：m 代表方程的个数；n 代表决策变量的个数。

进化算法在求解非线性方程组时需要将非线性方程组转换为优化问题。当前转换的方法主要单目标优化、约束优化和多目标优化有三种。第一种是将非线性方程组转换为单目标优化问题，它的形式如下：

$$\min\ P(x)=\sum_{i=1}^{m}\left|f_i(x_1,x_2,x_3,\dots,x_n)\right| \tag{2}$$

或者

$$\min\ P(x)=\sum_{i=1}^{m}f_i^2(x_1,x_2,x_3,\dots,x_n) \tag{3}$$

第二种是将非线性方程组转换为约束优化问题，它的形式如下：

$$\begin{cases} \min\ P(x)=\sum_{i=1}^{m}\left|f_i(x_1,x_2,x_3,\dots,x_n)\right| \\ \text{subject to}\ \ f_i(x_1,x_2,x_3,\dots,x_n)\ \geq 0,\ i=1,2,\dots,m \end{cases} \tag{4}$$

这两种转换方法的共同点是都有一个优化目标，得出的结果是一个最优解。而要想同时找到多个最优解，需要采用第三种转换方法：多目标优化。当前将非线性方程组转换为多目标优化问题的形式主要有三类，第一类是直接转换法，它的表达式如下：

$$\begin{cases} \min\ g_1(x)=\left|f_1(x_1,x_2,x_3,\dots,x_n)\right| \\ \min\ g_2(x)=\left|f_2(x_1,x_2,x_3,\dots,x_n)\right| \\ \qquad\vdots \\ \min\ g_m(x)=\left|f_m(x_1,x_2,x_3,\dots,x_n)\right| \end{cases} \tag{5}$$

该方法将方程组中每一个方程都当成一个优化目标，简单方便。但随着方程个数的增多，求解的复杂度将会增大。因此该方法只适用于具有少量方程的非线性方程组的求解。第二类方法是双目标转换法，它的表达式如下：

$$\begin{cases} \min\ f_1(x)=x_1+\sum_{i=1}^{m}\left|f_i(x_1,x_2,x_3,\dots,x_n)\right| \\ \min\ f_2(x)=1-x_1+\sum_{i=1}^{m}\left|f_i(x_1,x_2,x_3,\dots,x_n)\right| \end{cases} \tag{6}$$

该方法将线性方程与单目标优化方程结合，把非线性方程组的解转换成线性 Pareto 前沿上的解，直观有效。但是由于选择机制的作用容易造成部分解的丢失。第三类方法是多权值双目标转换法，它的表达式如下：

$$\begin{cases} \min\ f_1(x)=\dfrac{\sum_{i=1}^{n}\omega_i x_i}{\sum_{i=1}^{n}\omega_i}+\sum_{i=1}^{m}\left|f_i(x_1,x_2,x_3,\dots,x_n)\right| \\ \min\ f_2(x)=1-\dfrac{\sum_{i=1}^{n}\omega_i x_i}{\sum_{i=1}^{n}\omega_i}+\sum_{i=1}^{m}\left|f_i(x_1,x_2,x_3,\dots,x_n)\right| \end{cases} \tag{7}$$

该方法在双目标转换法的基础上对每一个未知变量都设定一个权值，能够有效避免最优解的丢失。因此，本文采用多权值双目标转换法求解非线性方程组。

2 标准差分进化算法

差分进化算法是一门新兴的进化计算技术，主要用于求解连续变量的全局优化问题。该算法通过变异、交叉、选择三种操作不断地迭代计算，保留优良个体，淘汰劣质个体，直到找到全局最优解。差分进化算法具有操作简单、控制参数少、易编程实现等优点，同时它又尤其适合求解多变量、非凸、多峰以及非线性优化问题。其基本的算法流程如下：

a)初始化种群个体和相关参数。主要包括种群大小、交叉率、缩放因子、最大评价次数和终止条件等。

b)变异。通过变异策略将当前个体进行变异。具体操作是通过把种群中两个成员之间的加权差向量加到第三个成员上，以此来产生新参数向量，从而产生新个体。

c)交叉。变异向量的参数与另外预先确定的目标向量参数按一定规则混合来产生实验向量。

d)选择。若实验向量的适应度值比目标向量的适应度值好，实验向量就在下一代中代替目标向量；否则，保留目标向量。对每一个体都进行选择以此得到新种群。

e)循环。直到满足终止条件，停止。

3 多模态多目标差分进化算法

多模态多目标优换问题，是指在决策空间中至少有两个

Pareto 解集对应目标空间上的同一个 Pareto 前沿。如图 1 所示，决策空间上的 Pareto 解集 1 和 Pareto 解集 2 同时对应目标空间上的同一条 Pareto 前沿。

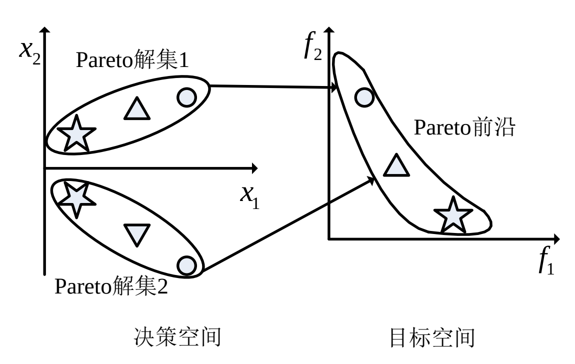


图 1　多模态多目标问题

而非线性方程组中也有一些解的分布情况与多模态多目标问题中的情况类似。基于这种思想，本文提出了一种多模态多目标差分进化算法（MMODE）。MMODE 在标准差分进化算法基础上又增加了一种个体预选择机制和一个变异边界处理方法，这样可以有效提高搜索精度和效率。它的具体步骤如下：

a)初始化。生成一个种群大小为 N 的随机种群 P，将缩放因子、交叉率、最大评价次数等赋初始值。

b)评价。利用评价函数评价 P 中每一个个体。

c)预选择机制。在目标空间中使用非支配解排序和决策空间中使用拥挤距离选择方法，选择前 50%的优良个体并将其放入 Q 中。

d)变异。对 Q 中的每一个个体采用 $DE/rand/2$ 策略进行变异。解决变异边界问题的方法具体参见 3.2 节。

e)交叉。利用二进制交叉算子产生子种群 R。

f)选择。在目标空间中使用非支配解排序和拥挤距离选择机制，选择 N 个个体进入下一代。

g)循环。如果满足终止条件，停止并返回全部最优解，否则重新返回步骤 c)。

3.1　个体预选择机制

在变异之前，通过设计一种个体预选择机制，将初始种群进行预处理，优先挑选 Pareto 最优前沿和决策空间中拥挤距离较大的解，这样可以有效增加所得解的多样性。预选择机制的过程如图 2 所示。首先将种群 P 中每个个体的适应度值经过非支配解排序得出排名 F1、F2、F3 等，颜色越深代表 Pareto 前沿排序越靠前；接着利用拥挤距离选择机制保留临界个体并舍弃拥挤距离小的个体；然后得到种群 Q，并利用种群 Q 进行变异。

3.2　变异边界处理方法

为了提高种群多样性并保证足够的搜索范围，本文采用 $DE/rand/2$ 进行变异。其表达式为

$$V_{i,G}=X_{r_1^i,G}+F*(X_{r_2^i,G}-X_{r_3^i,G})+F*(X_{r_4^i,G}-X_{r_5^i,G}) \tag{8}$$

其中：$X_{r_1^i,G}$、$X_{r_2^i,G}$、$X_{r_3^i,G}$、$X_{r_4^i,G}$ 和 $X_{r_5^i,G}$ 是从 $[1,N]$ 中随机生成的整数，且 $r_1^i \neq r_2^i \neq r_3^i \neq r_4^i \neq r_5^i \neq i$。缩放因子 F 的取值是 $[0,1]$。

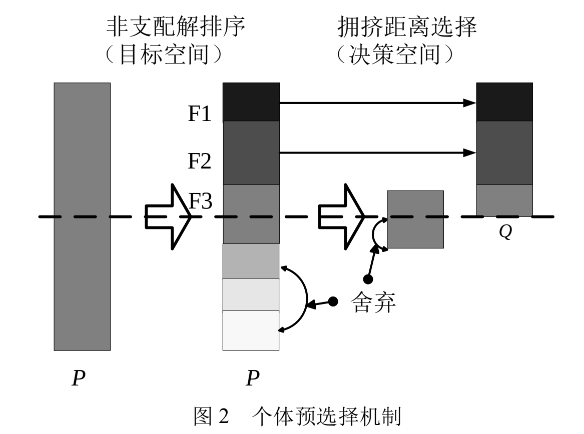
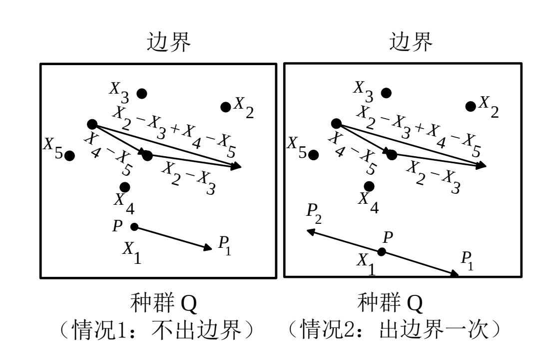


图 2　个体预选择机制

就传统差分进化算法而言，对于超出了搜索边界的个体往往将其置为边界值。而对于 MMODE，该算法加入了一种新的边界处理方法。它将停留在边界上的个体重新分布在整个决策空间中，目的是在搜索过程中保证解的多样性。这种新的变异策略表达式如下：

$$V_{i,G}=X_{r_1^i,G}-F*(X_{r_2^i,G}-X_{r_3^i,G})-F*(X_{r_4^i,G}-X_{r_5^i,G}) \tag{9}$$

若采用这种变异策略变异后新个体仍然超出边界，就将该个体随机分布在种群中。变异过程中的个体有三种情况，如图 3 所示。对于情况 1，因为不出边界，直接采用式（8）进行变异，变异前个体 P 变异成个体 P_1；对于情况 2，首先采用式（8）进行一次变异，所得变异个体超出边界，此时利用变异策略式（9）重新进行二次变异，新变异个体没有超出边界，所以个体 P 变异成个体 P_1 后再次变异成个体 P_2；对于情况 3，二次变异之后仍然超出边界，将个体重置为种群内随机个体，因此个体 P 变异成个体 P_1 再次变异成个体 P_2，最后设置为种群中的随机个体 O。



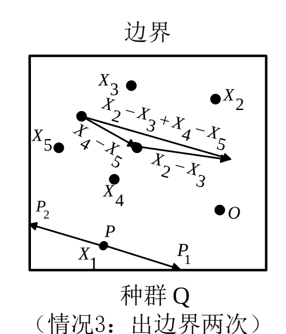


图 3　变异边界的三种情况和处理方法

4　实验及结果

4.1　测试函数描述

为了测试 MMODE 的有效性，选择文献[12]中的五个测试函数进行测试，分别是 F12、F15、F19、F27、F38。这五个测试函数解的个数如表 1 所示。这五个测试函数的数学表达式和解的具体信息如下：

表 1　被测试的五个非线性方程组

函数	F12	F15	F19	F27	F38
解	10	16	4	12	4

问题 1（F12）：

$$\begin{cases} \sin(x_1^3) - 3x_1x_2^2 - 1 = 0 \\ \cos(3x_1^2x_2) - |x_2^3| + 1 = 0 \end{cases} \tag{10}$$

其中：$x_i \in [-2,2], i = 1,\dots,n, n = 2$。该方程组有 10 个解，分别是 $(-1.81089,-0.34909)$，$(-1.81089,0.34909)$，$(-1.50222,-0.40908)$，$(-1.50222,0.40908)$，$(-1.79130,0.30193)$，$(-1.79130,-0.30193)$，$(-0.94727,0.78502)$，$(-0.94727,-0.78502)$，$(-0.21306,1.25685)$，$(-0.21306,-1.25685)$。

问题 2（F15）：

$$\begin{cases} x_1^2 + x_2^2 - 1.0 = 0 \\ x_3^2 + x_4^2 - 1.0 = 0 \\ x_5^2 + x_6^2 - 1.0 = 0 \\ x_7^2 + x_8^2 - 1.0 = 0 \\ 4.731*10^{-3}x_1x_3 - 0.3578x_2x_3 - 0.1238x_1 + x_7 \\ \quad -1.637*10^{-3}x_2 - 0.9338x_4 - 0.3571 = 0 \\ 0.2238x_1x_3 + 0.7623x_2x_3 + 0.2638x_1 - x_7 \\ \quad -0.07745x_2 - 0.6734x_4 - 0.6022 = 0 \\ x_6x_8 + 0.3578x_1 + 4.731*10^{-3}x_2 = 0 \\ -0.7623x_1 + 0.2238x_2 + 0.3461 = 0 \end{cases} \tag{11}$$

其中：$x_i \in [-1,1], i = 1,\dots,n, n = 8$。该方程组有 16 个解，分别是

$(0.1644,-0.9864,-0.9471,-0.321,-0.9982,-0.0594,0.411,0.9116)$，
$(0.1644,-0.9864,-0.9471,-0.321,-0.9982,0.0594,0.411,-0.9116)$，
$(0.1644,-0.9864,-0.9471,-0.321,0.9982,-0.0594,0.411,0.9116)$，
$(0.1644,-0.9864,-0.9471,-0.321,0.9982,0.0594,0.411,-0.9116)$，
$(0.1644,-0.9864,0.7185,-0.6956,-0.998,-0.0638,-0.5278,0.8494)$，
$(0.1644,-0.9864,0.7185,-0.6956,-0.998,0.0638,-0.5278,-0.8494)$，
$(0.1644,-0.9864,0.7185,-0.6956,0.998,-0.0638,-0.5278,0.8494)$，
$(0.1644,-0.9864,0.7185,-0.6956,0.998,0.0638,-0.5278,-0.8494)$，
$(0.6716,0.741,-0.6516,-0.7586,-0.9625,-0.2711,-0.4376,0.8992)$，
$(0.6716,0.741,-0.6516,-0.7586,-0.9625,0.2711,-0.4376,-0.8992)$，
$(0.6716,0.741,-0.6516,-0.7586,0.9625,-0.2711,-0.4376,0.8992)$，
$(0.6716,0.741,-0.6516,-0.7586,0.9625,0.2711,-0.4376,-0.8992)$，
$(0.6716,0.741,0.9519,-0.3064,-0.9638,-0.2666,0.4046,0.9145)$，
$(0.6716,0.741,0.9519,-0.3064,-0.9638,0.2666,0.4046,-0.9145)$，
$(0.6716,0.741,0.9519,-0.3064,0.9638,0.2666,0.4046,-0.9145)$，
$(0.6716,0.741,0.9519,-0.3064,0.9638,-0.2666,0.4046,0.9145)$。

问题 3（F19）：

$$\begin{cases} x_1^2 + x_2^2 - 2 = 0 \\ x_1^2 + x_2^2 / 4 - 1 = 0 \end{cases} \tag{12}$$

其中：$x_i \in [-2,2], i = 1,\dots,n, n = 2$。该方程组有 4 个解，分别是 $(-0.8165,-1.1547)$，$(0.8165,-1.1547)$，$(-0.8165,1.1547)$，$(0.8165,1.1547)$。

问题 4（F27）：

$$\begin{cases} 5x_1^9 - 6x_1^5x_2^2 + x_1x_2^4 + 2x_1x_3 = 0 \\ -2x_1^6x_2 + 2x_1^2x_2^3 + 2x_2x_3 = 0 \\ x_1^2 + x_2^2 - 0.265625 = 0 \end{cases} \tag{13}$$

其中 $x_1 \in [-0.6,6], x_2 \in [-0.6,0.6], x_3 \in [-5,5]$。该方程组有 12 个解，分别是

$(0.27986,0.43279,-0.01419)$，$(0.27986,-0.43279,-0.01419)$，
$(-0.27986,0.43279,-0.01419)$，$(-0.27986,-0.43279,-0.01419)$，
$(0.46698,0.21807,0.00000)$，$(-0.46698,0.21807,0.00000)$，
$(0.46698,-0.21807,0.00000)$，$(-0.46698,-0.21807,0.00000)$，
$(0.00000,0.51539,0.00000)$，$(0.00000,-0.51539,0.00000)$，
$(0.51539,0.00000,-0.01245)$，$(-0.51539,0.00000,-0.01245)$。

问题 5（F38）：

$$\begin{cases} x_1^4 + 4x_2^4 - 6.0 = 0 \\ x_1^2x_2 - 0.6787 = 0 \end{cases} \tag{14}$$

其中：$x_1 \in [-2,2], x_2 \in [0,1.1]$。该方程组有 4 个解，分别是 $(-1.56353,0.27763)$，$(-0.78971,1.08830)$，$(1.56353,0.27763)$，$(0.78971,1.08830)$。

4.2　评价指标

对于有限解的非线性方程组，需要在算法运行结束后确定已搜索到解的个数。以其中一个解为例，如果算法获得的解与已知最优解的欧氏距离小于预先设定的精确度 ε，就认为该最优解已经找到。这里将精确度设为

$$\varepsilon = \begin{cases} 0.01 & n \le 5 \\ 0.1 & \text{其他} \end{cases} \tag{15}$$

其中：n 为未知数的个数。算法获得的解的个数与所有最优解的个数之间的比值称为峰比率，用 PR 表示，PR 代表单次运行中找到最优解的平均概率。如果全部最优解都能在一次运行中找到，那么就称为一次成功的运行。成功运行的次数与全部运行总次数之间的比值称为成功率，用 SR 表示，SR 代表单次运行成功找到全部最优解的平均概率。本文采用 PR 和 SR 两个性能指标来评价当前算法的性能，它们的值越大，就说明算法的性能越好。

4.3　比较算法

与本文所提出的 MMODE 比较的算法如下：

a)A-WeB。该算法由 Gong 等人[12]于 2017 年提出，主要是用多权值双目标转换技术求解非线性方程组的最优解。

b)DN-NSGAII。该算法由 Yue 等人[13]于 2016 年提出，主要是用决策空间非支配解排序算法求解多模态多目标优化问题。

c)NSGAII。该算法由 Deb 等人[14]于 2002 年提出，用于求解非支配解排序的问题。

d)Omni-Opt。该算法由 Deb 等人[15]于 2005 年提出，用于改

善搜索解在决策空间的分布。

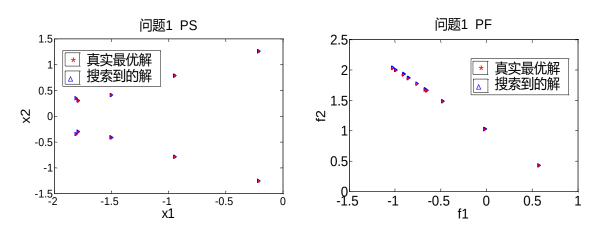
4.4 参数设置

在实验中参数设置如下：种群大小 NP=100，最大评价次数 $\max FES$=50000，缩放因子 F=0.5，交叉率 Cr=0.9。为保证比较的公平性，所有算法的种群大小、最大评价次数等参数均保持相等。每种算法均独立运行 51 次，结果取平均值。

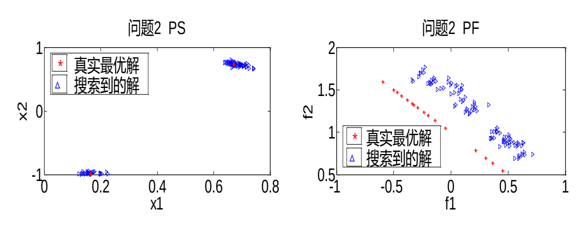
4.5 实验结果和分析

为避免偶然性，首先将第 26 次所提 MMODE 算法搜索到的非线性方程组的解的情况输出，其中问题 1～5 的所有解在决策空间和目标空间的分布如图 4 所示。从图 4（a）可以看出，对于有 2 个决策变量的问题 1，在决策空间和目标空间上，10 个解都能够找到，而且搜索到的解与真实最优解之间比较接近，可以看出 MMODE 对于问题 1 解决的比较好，它的 10 个最优解全部都能搜索到；从（b）可以看出，对于有 8 个决策变量的问题 2，由于维度的限制，这里给出了前两维的图形，在决策空间上，搜索到的解都分布在 16 个最优解的周围，而在目标空间上，搜索到的解与真实最优解之间还有一定的差距，可以看出 MMODE 对于问题 2 解决的并不好，它的 16 个最优解还不能很好的搜索到，说明 MMODE 对于决策变量较多的问题 2 的求解上仍然还有待改进；从（c）可以看出，对于有 2 个决策变量的问题 3，在决策空间和目标空间上，4 个解都能够找到，而且搜索到的解与真实最优解之间比较接近，不过仍存在一些与真实最优解相距较远的搜索解，可以看出 MMODE 对于问题 3 解决的比较好，它的 4 个最优解全部都能搜索到，但是搜索到的全部解还不能完全地趋近真实最优解；从（d）可以看出，对于有 3 个决策变量的问题 4，在决策空间和目标空间上，12 个真实最优解只有 8 个能够被找到，另外 4 个最优解与搜索到的解还有一些差距，可以看出 MMODE 对于问题 4 只能搜索到一部分真实最优解；从（e）可以看出，对于有 2 个决策变量的问题 5，在决策空间和目标空间上，4 个真实最优解都能被很好地找到，可以看出 MMODE 对于问题 5 的求解表现的比较好。

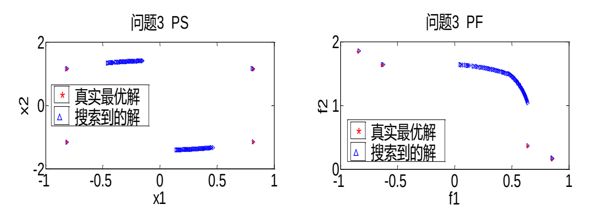
接下来，将 MMODE 算法与 A-WeB、DN-NSGAII、NSGAII 和 Omni-Opt 算法在峰值率上进行比较，详细结果如表 2 所示。从表 2 可以看出，对于问题 1，MMODE 在峰比率 PR 上数值最大，A-WeB 排第二，　A-WeB 和 MMODE 相差不大,NSGAII、Omni-Opt 和 DN-NSGAII 分别排在第三到第五，说明 MMODE 平均搜索到解的个数比其他算法更多；对于问题 2，MMODE 在峰比率 PR 上数值最大，A-WeB 排在第二，NSGAII、DN-NSGAII 和 Omni-Opt 分别排在第三到第五，与前两名相差较多，说明 MMODE 平均搜索到的解的个数比其他算法更多，并且效果比其他算法表现更好；对于问题 3，MMODE 在峰比率 PR 上数值为 1，A-WeB、NSGAII、DN-NSGAII 和 Omni-Opt 分别排在第二到第五，但与 MMODE 相差不大，说明 MMODE 每次都能够搜索到全部最优解，其他四种算法也可以搜索到绝大部分的最优解；对于问题 4，MMODE 在峰比率 PR 上数值最大，Omni-Opt、DN-NSGAII、NSGAII 和 A-WeB 和分别排在第二到第五，说明 MMODE 在五个算法比较上表现最好，但是全部算法平均搜索到的最优解并不多；对于问题 5，MMODE 和 A-WeB 在峰比率 PR 上数值均为 1，NSGAII 排在中间，DN-NSGAII 和 Omni-Opt 的 PR 值一样大，排在最后，说明 MMODE 和 A-WeB 每一次都能够搜索到全部最优解，其他三个算法可以搜索到大部分的最优解。



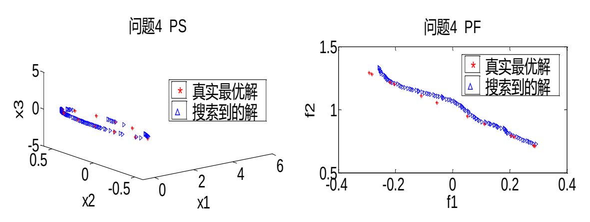
(a)问题 1 的 10 个最优解在决策空间和目标空间的分布



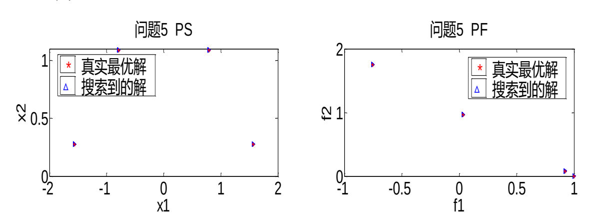
(b)问题 2 的 16 个最优解在决策空间和目标空间的分布



(c)问题 3 的 4 个最优解在决策空间和目标空间的分布



(d)问题 4 的 12 个最优解在决策空间和目标空间的分布



(e)问题 5 的 4 个最优解在决策空间和目标空间的分布

图 4　问题 1～5 的所有解在决策空间和目标空间的分布

表 2　MMODE 和四个算法在问题 1-5 上的峰比率 PR

算法	问题 1	问题 2	问题 3	问题 4	问题 5
MMODE	**0.8980**	**0.7350**	**1.0000**	**0.3967**	**1.0000**
A-WeB	0.8880	0.6688	0.9950	0.0933	**1.0000**
DN-NSGAII	0.4080	0.0850	0.9650	0.2800	0.7100
NSGAII	0.6400	0.0888	0.9800	0.2783	0.7250
Omni-Opt	0.5020	0.0725	0.9550	0.2900	0.7100

为了避免单独比较峰值率造成的偏差，现在将五种算法在成功率上进行比较，详细结果如表 3 所示。从表 3 可以看出，对于问题 1，MMODE 和 A-WeB 在成功率 *SR* 上数值一样，排在第一，DN-NSGAII、NSGAII 和 Omni-Opt 数值为零，说明 MMODE 和 A-WeB 成功找到全部最优解的次数最多，但是成功率并不高；对于问题 2，MMODE 排在第一，其他四种算法成功率均为零，说明 MMODE 可以找到全部最优解，但是成功率非常低；对于问题 3，MMODE 成功率是 1，A-WeB、NSGAII、DN-NSGAII 和 Omni-Opt 分别排在第二到第五，说明 MMODE 每次运行都可以成功找到全部最优解；对于问题 4，A-WeB 成功率排在第一，MMODE、DN-NSGAII、NSGAII 和 Omni-Opt 成功率均为零，说明 MMODE 每次运行都不能成功找到全部最优解；对于问题 5，MMODE 和 A-WeB 排在第一，并且成功率都是 1，NSGAII、DN-NSGAII 和 Omni-Opt 依次排在后三位，与前两名相差较大，说明 MMODE 和 A-WeB 每次运行都能成功找到全部最优解。

表 3　MMODE 和四个算法在问题 1-5 上的成功率 *SR*

算法	问题 1	问题 2	问题 3	问题 4	问题 5
MMODE	**0.2800**	**0.0400**	**1.0000**	0.0000	**1.0000**
A-WeB	0.2800	0.0000	0.9800	**0.0933**	**1.0000**
DN-NSGAII	0.0000	0.0000	0.8600	0.0000	0.0600
NSGAII	0.0000	0.0000	0.9200	0.0000	0.0800
Omni-Opt	0.0000	0.0000	0.8200	0.0000	0.0400

综合表 2 和 3 可以看出，对于大部分测试问题，MMODE 在峰比率和成功率上的结果比其他四种算法表现更好，说明 MMODE 在求解非线性方程组问题上是有效的。

另外，为了更好地观察实验数据的分布，对其中的四种算法的峰值率 *PR* 所对应的盒形图表示如图 5 所示。其中横坐标 1、2、3、4 分别代表 MMODE、DN-NSGAII、NSGAII、Omni-Opt 四种算法，纵坐标代表峰值率 *PR* 值。从图 5 可以看出，对于问题 1～4，MMODE 和其他三种算法的波动范围大致相同，但峰值率的均值比其他三种算法更大；对于问题 3，四种算法的峰值率波动性比较小，峰值率都接近于 1，但是 MMODE 比其他三种算法更加稳定，每次都可以找到全部最优解；对于问题 5，四种算法的峰值率波动性也比较小，但 MMODE 的峰值率可以一直保持为 1，这比其他三种算法长期保持在 0.7 附近效果更好。因此，从盒形图上看，MMODE 在大部分问题上比其他算法表现更好。

为了进一步比较 MMODE 算法对其他四种算法的优势，这里使用 Wilcoxon 秩和检验进行分析。因为考虑到搜索到的非线性方程组解的平均数量是当前判断算法好坏的重要标志，所以这里检验的对象是 MMODE 和其他四种算法的全部 51 次峰值率。通过进行 Wilcoxon 秩和检验，比较结果如表 4 所示，其中全部检验结果都是在显著性水平为 0.05 下进行的。从表 4 可以看出，与 MMODE 比较的四种算法得到的 *P* 值都远远小于 0.05，MMODE 的显著性优势从小到大依次是 A-WeB、NSGAII、Omni-Opt 和 DN-NSGAII，说明 MMODE 算法显著性地优于其他四种算法。

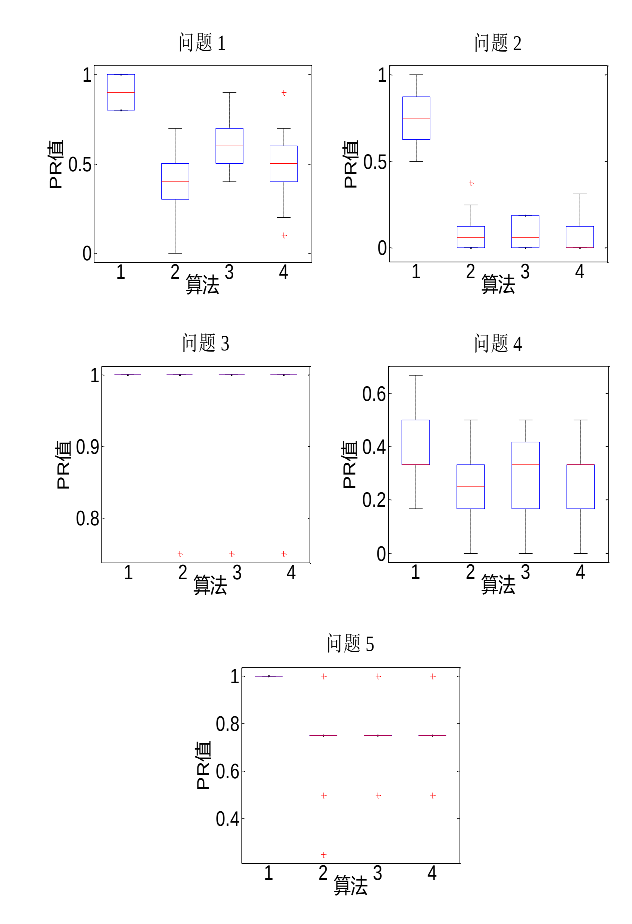


图 5　四种算法的盒形图

表 4　MMODE 和四个算法在问题 1～5 上的 Wilcoxon 秩和检验结果

MMODE　vs	*P* 值
A-WeB	3.7429e-13
DN-NSGAII	6.0879e-25
NSGAII	4.4421e-19
Omni-Opt	4.4891e-24

5　应用

非线性方程组在工程中也有广泛的应用，比如可以用来建立两个连续的非绝缘搅拌釜式反应器的数学模型，从而得出系统稳定后两个反应器的温度[16]。这些反应往往处在一个稳定可循环的状态，并且是一个一阶不可逆的放热反应。忽略个别因素的影响，该反应器的数学模型构建如下：

$$\begin{cases} f_1 = (1-R)\left[\dfrac{D}{10(1+\beta_1)} - x_1\right]\exp(\dfrac{10x_1}{1+10x_1/\gamma}) - x_1 \\ f_2 = x_1 - (1+\beta_2)x_2 + (1-R)\left[\dfrac{D}{10} - \beta_1 x_1 - (1+\beta_2)x_2\right]\exp(\dfrac{10x_2}{1+10x_2/\gamma}) \end{cases} \quad (16)$$

其中：x_1 和 x_2 分别代表两个反应器中的无量纲温度；γ 表示无量纲激活能量，$\gamma = 1000$；D 表示达姆科勒数，$D = 22$；β_1 和 β_2 分别表示无量纲传热系数，$\beta_1 = \beta_2 = 2$；R 表示循环率，$R = 0.96$。

现将 MMODE 和四种算法分别用来求解这一工程实例，评价指标峰比率和成功率的结果如表 5 所示。

表 5　MMODE 和四个算法在工程实例上的结果

评价指标	MMODE	A-WeB	DN-NSGAII	NSGAII	Omni-Opt
峰比率	**0.8571**	0.8371	0.8367	0.8385	0.8543
成功率	**0.1569**	0.1176	0.0784	0.0784	0.0980

从表 5 可以看出，在峰比率上，五种算法的结果总体相差较小，MMODE 峰比率最大；在成功率上，五种算法的成功率都不高，但相比较而言，MMODE 有最大的成功率。综合比较峰比率和成功率的结果可以得出，MMODE 在非绝缘搅拌釜式反应器上的应用也比其他四种算法具有更好的有效性。

6　结束语

本文主要介绍了非线性方程组的求解，首先总结了将非线性方程组转换为优化问题的主要方法；然后根据当前算法存在的问题，提出了一种多模态多目标差分进化算法。该算法的主要改进包括种群预选择机制和边界选取方法，这些改进能够增加决策空间解的多样性。为了验证多模态多目标差分进化算法的有效性，对 5 个测试函数进行了 51 次独立实验。最后经过与四种算法得出的结果进行比较，可以说明多模态多目标差分进化算法在峰比率和成功率两个指标上都有一定的优势。另外，通过求解一个具体的工程实例，也进一步验证了所提算法的有效性。因此，多模态多目标差分进化算法是一种有效求解非线性方程组的方法。不过，对于非线性方程组问题的求解，尤其是在成功率上，当前算法仍然还有很大的提升空间。下一步工作将继续优化算法，使之能够更好地应用到更多复杂的工程实例中。

参考文献：

[1] 陶征，胡斌，刘旭．基于螺栓联接非线性系统的杆式超声电机建模与分析 [J]. 郑州大学学报：工学版, 2017, 38 (1): 78-82.

[2] 沈培辉，林述温．双钢轮压路机的非线性建模及其传动干扰分析 [J]. 郑州大学学报：工学版, 2017, 38 (2): 66-71.

[3] 罗亚中，袁端才，唐国金．求解非线性方程组的混合遗传算法 [J]. 计算机力学学报, 2005, 22 (1): 109-114.

[4] 林彬，王正峰．基于差分进化算法求解非线性方程组 [J]. 环球市场信息导报, 2016, 2016 (9): 78-78.

[5] 王冬冬，周永权．人工鱼群算法在求解非线性方程组中的应用 [J]. 计算机应用研究, 2007, 24 (6): 242-244.

[6] 汪继文，杨丹，邱剑锋，等．改进人工蜂群算法求解非线性方程组 [J]. 安徽大学学报：自然科学版, 2014, 2014 (3): 16-23.

[7] 赵光伟，周永权．用改进的人工萤火虫群优化算法求解非线性方程组 [J]. 数学的实践与认识, 2016, 46 (1): 176-186.

[8] 樊晓敏，许天亮．求解非线性方程组的改进布谷鸟算法 [J]. 内蒙古师大学报：自然汉文版, 2017, 46 (1): 13-16.

[9] 陈海霞，杨铁贵．基于极大熵差分进化混合算法求解非线性方程组 [J]. 计算机应用研究, 2010, 27 (6): 2028-2030.

[10] 刘利斌，欧阳艾嘉，许卫明，等．求解非线性方程组的 BFGS 差分进化算法 [J]. 计算机工程与应用, 2011, 47 (33): 55-58.

[11] 封全喜，刘三阳，唐国强，等．求解方程组的正交差分进化算法 [J]. 计算机科学, 2012, 39 (5): 187-189.

[12] Wenyin Gong, Yong Wang, Zhihua Cai, *et al.* A weighted biobjective transformation technique for locating multiple optimal solutions of nonlinear equation systems [J]. IEEE Trans on Evolutionary Computation, 2017, 21 (5): 697-713.

[13] Caitong Yue, Jing Liang, Boyang Qu. Multimodal multi-objective optimization: a preliminary study [C]// Proc of IEEE Congress on Evolutionary Computation. 2016: 2454-2461.

[14] Deb K, Pratap A, Agarwal S, *et al.* A fast and elitist multi-objective genetic algorithm: NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2002, 6 (2): 182-197.

[15] Deb K, Tiwari S. Omni-optimizer: a procedure for single and multi-objective optimization [J]. Evolutionary Multi-Criterion Optimization, 2005, 3410: 47-61.

[16] Hirsch M J, Pardalos P M, Resende M G C. Solving systems of nonlinear equations with continuous GRASP [J]. Nonlinear Analysis Real World Applications, 2009, 10 (4): 2000-2006.